UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

TRABAJO FIN DE GRADO

Detección de intrusión con cámaras móviles en tiempo real

Alberto Palero Almazán

MAYO 2014

Detección de intrusión con cámaras móviles en tiempo real

AUTOR: Alberto Palero Almazán

TUTOR: Jesús Bescós Cano

Grupo VPULab

Dpto. de Tecnología Electrónica y de las Comunicaciones

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Mayo de 2014

Trabajo parcialmente financiado por el gobierno español bajo el proyecto TEC2011-25995 (EventVideo)

Agradecimientos

Quiero agradecer a mi tutor, Jesús Bescós, por la oportunidad de poder realizar este trabajo. Sin su ayuda y su apoyo, aquí y a lo largo de toda la carrera, esto no hubiese sido posible.

También agradecer a José María Martínez por su ayuda y enseñanzas, sin él este día tampoco hubiese llegado.

Gracias a toda la gente del VPU por los consejos y la ayuda, en especial a Carlos por siempre estar dispuesta a echar una mano y a Sara, Patricia y Alejandro por su apoyo y por todas las risas que hacen que todo sea más fácil.

A todo el personal de la biblioteca de la EPS que hicieron que el tiempo que estuve allí me lo pasase genial. En especial a Isa y Ana, gracias por todos los buenos momentos.

A todos los compañeros de clase y de prácticas porque este trabajo es en parte gracias a todos vosotros.

No se expresar con palabras lo agradecido que estoy a Alejandro, Álvaro, Manuel y Adrián por todos estos años y siempre estar ahí para lo que fuese. Os Estaré eternamente agradecido.

Finalmente, quiero agradecer a las personas más importantes para mí, mis padres mi hermana y toda mi familia, por todo su apoyo y todo su cariño, por siempre creer en mí y por esforzarse para que llegase a donde estoy ahora. Nunca os podré devolver lo que me habéis dado.

Alberto Palero Almazán

Mayo 2014

PALABRAS CLAVE

Panorámica, segmentador frente-fondo, homografía, puntos SURF, cámara PTZ, vídeo-vigilancia.

RESUMEN

La motivación principal detrás de este trabajo ha sido desarrollar un sistema de detección de intrusos en un entorno de vídeo-vigilancia, que utilice un segmentador frente-fondo haciendo uso de una cámara móvil (cámara PTZ) de forma que se aumente con ella el campo de visión abarcado, comparado con una cámara fija.

Un segmentador frente-fondo está en general diseñado para cámaras fijas. Para adaptarlo a una cámara PTZ se propone alimentarlo con imágenes panorámicas de la escena que abarca la cámara.

En este trabajo se detalla el proceso llevado a cabo para la creación de la imagen panorámica que se utilizará como imagen de fondo en el segmentador, al igual que todos los problemas que pueden surgir en este proceso y sus respectivas soluciones propuestas.

Una vez generada la imagen panorámica de fondo, se explica cómo combinarla junto a la imagen procedente de la cámara PTZ en cada instante, de manera que ésta se inserte en la imagen panorámica y utilizar esta imagen panorámica con el *frame* de la cámara insertado en ella como entrada del segmentador. Finalmente, es el propio segmentador el encargado de marcar sobre qué zonas de la imagen panorámica hay intrusión.

Por último se ha evaluado, tanto el algoritmo generador de panorámicas como la solución completa propuesta, incluyendo el segmentador. Con ello se han sacado las conclusiones apropiadas.

KEYWORDS

Panorama, background subtraction, homography, SURF points, PTZ cameras, video surveillance.

ABSTRACT

The main motivation behind this project has been to develop an intrusion-detection system based on a video surveillance system that uses a background subtraction algorithm which uses a moving camera (PTZ camera) so that, compared with a fixed camera, it increases its field of view.

In general, a background subtraction algorithm is designed for fixed cameras. In order to adapt it to PTZ cameras we propose using panoramic pictures as the input for the background subtraction algorithm.

The process used to generate the panoramic image that will be used in the background subtraction algorithm is explained in this project, as well as all the problems that can be encountered in this process and all the solutions we propose to solve said problems.

Once the panoramic image has been generated, it is explained how to use this image and the new frame that comes from the PTZ camera, in each instance, so that these two images are merged together, and this new image, that is a mix between the front of the new frame and the panoramic image, as the input image of the background subtraction algorithm. The output will be a mask that overlaps over the panoramic image and highlights the areas of the image where there is intrusion.

Finally it has been evaluated separately, first the algorithm that creates the panoramic image and the complete solution, that includes the background subtraction. Conclusions have been obtained based on these results.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ECUACIONES

Capítulo 1: INTRODUCCIÓN

1.1 MOTIVACIÓN

En el área de análisis de secuencias de vídeo interesa segmentar o segregar partes de cada imagen para detectar distintas zonas de interés, ya sean personas u objetos con distinto significado dependiendo de la escena en la que aparecen.

Un tipo de algoritmo de segmentación tiene como objetivo, partiendo de las imágenes captadas por una cámara fija, dividir cada imagen en dos regiones: el frente, también llamado *foreground*, o zonas de la imagen donde hay movimiento; y el fondo, o *background*, que son las zonas estáticas de la imagen.

Actualmente, en el campo de la vídeo-vigilancia, los algoritmos de segmentación frente-fondo en general utilizan cámaras fijas lo que implica que se obtiene un campo de visión acotado, definido por la lente de la cámara. En este sentido, existen cámaras diseñadas con un campo de visión omnidireccional, mediante el uso de grandes angulares u ojos de pez, pero sujetas a grandes distorsiones radiales y, por lo tanto, a resoluciones variables en cada zona de la imagen

En este trabajo se utilizan cámaras PTZ (acrónimo de la expresión inglesa *pan-tilt-zoom*) que son cámaras ancladas en un punto fijo con respecto al cual pivotan en sentido horizontal (*pan*) y vertical (*tilt*). Comparadas con las cámaras omnidireccionales consiguen abarcar un amplio campo de visión, sin los problemas de distorsión y resolución variable, a costa de no poder captarlo completo en una única imagen correspondiente a un mismo instante.

Las citadas limitaciones que tienen las cámaras PTZ pueden ser resueltas utilizando técnicas existentes para generar imágenes panorámicas (una única imagen que va actualizándose para todo el campo de visión) y moviendo la cámara regularmente a cierta velocidad (de modo que el panorama completo corresponde a un intervalo de tiempo reducido). Para obtener un panorama a partir de varias imágenes es necesario calcular la transformación de cada imagen de forma que se forme una panorámica. Para ello, se necesitan calcular las homografías entre imágenes a partir de los puntos característicos de las imágenes, asunto éste resuelto en la actualidad. Para actualizar el panorama conforme se desplaza la cámara es posible plantear distintas estrategias.

La motivación principal de este trabajo es aumentar el campo de visión de la imagen utilizada como entrada al algoritmo de segmentación a partir de la secuencia cíclica de imágenes captadas desde una cámara PTZ en movimiento continuo.

1.2 OBJETIVOS

El objetivo principal de este trabajo es generar una imagen panorámica, a partir de varias imágenes recibidas de una cámara PTZ, de forma que el campo de visión aumente, y utilizar esta imagen como entrada para un algoritmo de *background subtraction*. Para poder realizar dicho objetivo, este trabajo se divide en las siguientes tareas:

- **Estudio del estado del arte:** Se analizarán diversos métodos de generación de panorámicas del estado del arte actual, prestando especial atención a aquellos métodos enfocados a cámaras PTZ.
- **Implementación del algoritmo de generación de panorámicas:** Se desarrollará un algoritmo que genere una imagen panorámica y que la actualice con cada barrido de la cámara.
- **Integración en un algoritmo de *background subtraction*:** Se analizarán estrategias para actualizar la panorámica en caso de cambios relevantes en la escena, para así poder utilizarla como entrada de algoritmos de análisis, en particular de *background subtraction*.
- **Evaluación del algoritmo generador de panorámicas:** Se evaluará la imagen generada de forma que se pueda medir cuantitativamente la calidad de dicha imagen panorámica.
- **Evaluación del algoritmo de segmentación:** Una vez evaluada la calidad de la imagen panorámica se evaluará los resultados de actualizar y utilizar dicha imagen como entrada del algoritmo de *background subtraction*.

1.3 ESTRUCTURA DE LA MEMORIA

La memoria está organizada de la siguiente manera:

- Capítulo 1: Motivación y objetivos del proyecto y estructura de la memoria.
- Capítulo 2: Estado del arte de algoritmos de *background subtraction* y su posible adaptación a cámaras PTZ. Identificación y breve descripción de las herramientas y entorno utilizados.
- Capítulo 3: Generación y actualización de la imagen panorámica de fondo a partir del cálculo de homografías. Problemas encontrados con este método y las soluciones propuestas.

- Capítulo 4: Descripción de la integración de la imagen panorámica en un algoritmo de *background subtraction*.
- Capítulo 5: Evaluación de la imagen panorámica generada y del algoritmo de segmentación completo.
- Capítulo 6: Conclusiones finales y posibles mejoras en un trabajo futuro.

Capítulo 2: ESTADO DEL ARTE Y CONCEPTOS BÁSICOS

Una vez que se ha enunciado la motivación y los objetivos principales del trabajo, se continúa con un estudio del estado del arte de algoritmos de *background subtraction* adaptados a las cámaras PTZ que son las que se van a utilizar.

Una vez hecho eso se hará una breve descripción de los conceptos básicos en los cuales se basa este trabajo.

2.1 ALGORITMOS DE *BACKGROUND SUBTRACTION*

Background subtraction es una técnica muy utilizada para detectar objetos en movimiento en videos donde la cámara está fija. El método más básico para hacer esto mantiene un llamado modelo de fondo, consistente en una imagen de la escena vacía, y resta el *frame* actual del modelo de fondo, para detectar las diferencias u objetos en movimiento. Este modelo se va actualizando para así adaptarse a condiciones variables de luminosidad o posibles cambios en los objetos que componen el fondo. Existen también modelos más complejos que llevan este principio básico más allá.

Un modelo simple pero más complejo que el método descrito en el párrafo anterior es el de Wren *et al.* [11]. Dicho algoritmo de *background subtraction* se basa en hacer un modelo independiente para cada píxel de la imagen, generando una función de densidad de probabilidad (pdf) gaussiana basada en el valor de los *n* píxeles homólogos de las *n* imágenes anteriores. La *resta* entre el *frame* actual y el modelo consiste en este caso en comprobar si cada pixel del *frame* pertenece o no a la pdf del píxel correspondiente del modelo. La media de cada pdf gaussiana se va actualizando por cada *frame* al igual que su desviación típica.

Otro método para obtener el modelo de fondo es el propuesto por Lo y Velastin en [12]. Ellos propusieron que se podía obtener el modelo de fondo asociando a cada píxel del modelo la mediana de los valores de los píxeles de las *n* imágenes anteriores. Esto hace posible que se gane una estabilidad del modelo de fondo a costa de necesitar un buffer con las *n* imágenes anteriores.

Además de estos métodos comentados existen muchos otros métodos de los cuales algunos de los más importantes están resumidos en [10].

2.1.1 Adaptación a cámaras PTZ

Como ya se ha mencionado, los algoritmos de *background subtraction* descritos anteriormente están diseñados para trabajar con una cámara fija, ya que si el modelo de fondo no corresponde al mismo punto de vista que la *frame* actual, la resta de ambos dará lugar en general a valores altos en todos los píxeles. Sin embargo las cámaras de vídeo-vigilancia, generalmente, son móviles pivotando desde un punto fijo con el principal objetivo de aumentar el campo de visión. Aunque no estén diseñados para este tipo de cámaras, los algoritmos de *background subtraction* pueden ser ligeramente modificados para poder seguir funcionando de la misma manera que utilizando una cámara fija.

En los últimos años se ha trabajado, con diversos enfoques, en adaptar los algoritmos de *background subtraction* a cámaras de tipo PTZ. Muchos de esos trabajos se desarrollan dado el modelo de fondo pero pocos se encargan de calcularlo previamente. De estos pocos, las aproximaciones más comunes tratan de simplificar las transformaciones proyectivas a afines o rígidas de 2 dimensiones [14], aprovechar la información específica de la cámara, como puede ser su posición [15] o generar el modelo sin información previa [4]. Otro método es el utilizado en [16], pero debido a que se necesita conocer la secuencia entera para poder corregir errores, hace que no sea posible llevarlo a cabo en tiempo real, que es el escenario típicamente requerido por un sistema de vídeo-vigilancia.

En el artículo de Bevilacqua [4] se consigue adaptar un algoritmo de *background subtraction* a una cámara PTZ de manera parecida al de este trabajo, con muy buenos resultados.

2.2 MÉTODOS PARA LA GENERACIÓN DE PANORÁMICAS

Tal y como se ha mencionado en la sección 1.2 el objetivo principal de este trabajo es generar una panorámica para poder así aumentar el campo de visión de una cámara PTZ, con intención de utilizar esta imagen panorámica como imagen de fondo para un algoritmo de *background subtraction*.

Para la generación de dicha panorámica hay dos herramientas esenciales: la localización de puntos homólogos (en la escena) entre dos vistas captadas por la cámara, y el cálculo de la homografía que mejor relaciona ambas vistas a partir de un número elevado de correspondencias de puntos.

Respecto de la primera, la mayor parte de las implementaciones disponibles hace uso de SURF [1] (acrónimo de la expresión inglesa *Speeded-Up Robust Features*)

como esquema robusto y eficiente de localización y caracterización de puntos de interés. A continuación se hará un breve descripción de que son, para que sirven y como se utilizan estas herramientas.

2.2.1 Puntos SURF

La técnica SURF extrae y describe zonas características de la imagen que son invariantes a rotaciones y a escala. Este método está basado en el de SIFT [18] y de la misma manera se divide en dos partes, el detector de puntos y el descriptor de la región en torno a ellos, pero están diseñados para que sean más rápidos de calcular sin sacrificar rendimiento. Estos puntos deben de ser muy precisos ya que son fundamentales para poder relacionar dos imágenes de la misma escena cambiando el punto de vista.

Para calcularlos puntos de interés de una imagen SURF se basa en localizar los máximos del determinante de la matriz Hessiana a cada escala. La matriz Hessiana es la siguiente:

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \quad (1)$$

, donde $L_{xx}(x,\sigma)$ es la segunda derivada parcial de una gaussiana en $\vec{x}$ de la imagen en el punto $\vec{x}$ = (x, y) a un escala σ. Para agilizar los cálculos se utilizan aproximaciones de las segundas derivadas de una gaussiana (ver Figura 2-1) de la imagen original mediante filtros de caja (*box filters*) además de *integral images* para obtener su resultado.

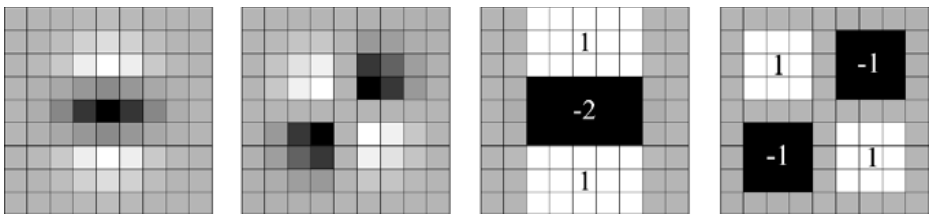


Figura 2-1: A la izquierda las derivadas parciales de segundo orden de una gaussiana, a la derecha las aproximaciones de SURF. Fuente: [1]

Las *integral images* son imágenes donde el valor de un píxel, *x* = (x, y), es igual a la suma de los píxeles dentro de la región rectangular desde el origen hasta el punto *x*.

Esto permite que, una vez calculada dicha imagen, sólo son necesarias tres sumas para obtener la suma de los valores de cualquier región rectangular de la imagen, independientemente de su tamaño, lo cual permite un cálculo rápido de un filtro de caja.

Los puntos de interés deben ser localizados a diferentes escalas, debido a que normalmente al buscar correspondencias entre puntos las imágenes pueden estar a diferentes escalas. Gracias a los filtros de caja y a las *integral images* se pueden aplicar filtros de caja de cualquier tamaño sin aumentar la velocidad, de forma que el espacio de escala se genera incrementando el tamaño de los filtros en vez de ir reduciendo el tamaño de la imagen (ver Figura 2-2).

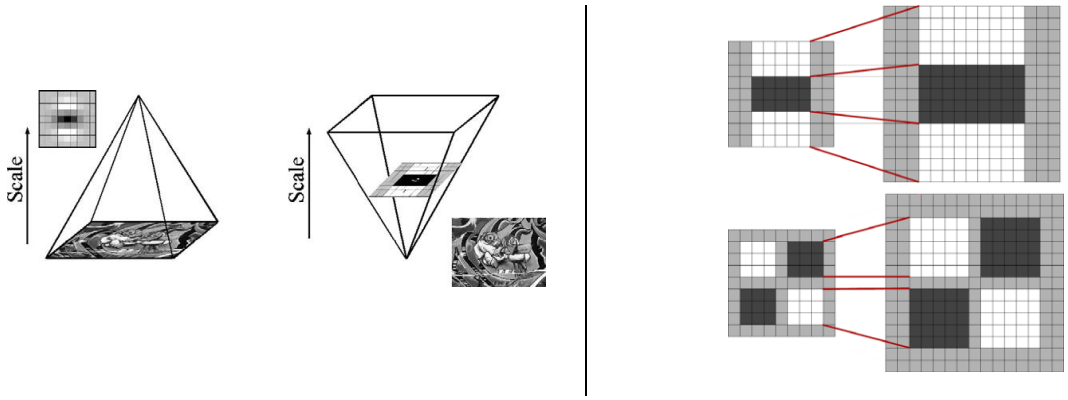


Figura 2-2: (a) En vez de reducir el tamaño de la imagen (izq.), se aumenta el tamaño del filtro (dcha.), (b) filtros para dos niveles de escala sucesivos, de 9x9 a 15x15. Fuente [1]

El detector de SURF también es muy parecido al de SIFT, describe la distribución de la intensidad del contenido en la región que rodea al punto de interés de manera similar a la información del gradiente extraído por SIFT. Para acelerar el proceso utiliza la distribución de respuestas de primer orden del *wavelet* de Haar e *integral images*.

2.2.2 Homografías

Las homografías o transformaciones proyectivas describen la transformación que experimenta un plano proyectado cuando la posición del observador, o de la cámara, varía. Es decir, si tenemos dos cámaras mirando a la misma escena plana, la relación entre las dos imágenes es una homografía. En el caso que nos ocupa, al estar la cámara PTZ anclada en un punto fijo, no hablamos de varias proyecciones sino de una única

proyección de la escena aunque captada apuntando a distintas direcciones. En este caso es posible demostrar que la relación entre dos vistas de la misma escena viene dada por una homografía con independencia de si la escena es o no plana.

La ecuación que relaciona las dos imágenes o vistas de la misma escena es la siguiente:

$$\begin{bmatrix} x' \\ y' \\ t' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{23} & h_{33} \end{bmatrix} * \begin{bmatrix} x \\ y \\ t \end{bmatrix} \quad (2)$$

, donde h_{33} se normaliza a uno, $\overrightarrow{x} = [x,\ y,\ t]$ son las coordenadas homogéneas de un pixel arbitrario en la imagen original que va a ser transformada y $\vec{x}' = [x',\ y',\ t']$ son las coordenadas homogéneas de dicho punto en la nueva imagen.

Para poder calcular la matriz que transforma la imagen es necesario obtener al menos cuatro correspondencias entre puntos x y x' no alineados de ambas imágenes, la que queremos transformar y a la que queremos que se transforme. Sin embargo, en la práctica, para conseguir mayor robustez en el cálculo de la homografía se obtiene un número elevado (decenas o incluso cientos) de correspondencias y se busca la homografía que mejor se adapta a todas ellas. La técnica para obtener los puntos característicos junto con sus descriptores ha sido SURF. Una vez aplicado SURF a ambas imágenes se hace un *matching* entre los puntos característicos de ambas imágenes, que es el resultado de aplicar un umbral a la distancia euclídea entre sus descriptores. Una vez obtenido el conjunto de puntos emparejados, los puntos cuyos descriptores presentan la menor distancia euclídea, se utiliza el algoritmo RANSAC [2] para obtener la homografía que minimiza un determinado criterio de error.

Las utilidades que tienen las homografías son muchas, entre las cuales se encuentran la generación de panorámicas como en este trabajo y en [3], [4] y [5], en algoritmos de seguimiento de personas u objetos como en [6], [7] y [8] y aplicaciones de calibración automática de cámaras como en [9], entre muchas otras aplicaciones.

2.3 LA PLATAFORMA DIVA

DiVA es la plataforma utilizada en este trabajo para la recepción en *streaming* de vídeo en directo desde varias fuentes disponibles en la Escuela Politécnica Superior de la Universidad Autónoma de Madrid.

La plataforma DiVA ha sido desarrollada por el VPU (*Video Processing and Understanding Lab*) de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid y permite establecer un entorno el cual permite obtener videos desde varias fuentes, comunicación con distintos algoritmos y procesamiento de manera secuencial o en paralelo, y todo ello en tiempo real.

El diseño de esta plataforma está basado en los siguientes criterios: escalabilidad, eficiencia, generalidad y tolerancia a errores. Es flexible de forma que permite soportar fuentes nuevas de video y protocolos, realiza operaciones de manera que el coste computacional no es elevado y es capaz de detectar y corregir errores mientras se está ejecutando.

Para integrar un algoritmo nuevo a la plataforma DiVA se sigue un proceso que está dividido en tres niveles (ver Figura 2-3) de los cuales el último es opcional. Para cada nivel, según el esquema de la figura 2-3, el algoritmo debe cumplir unos requisitos.

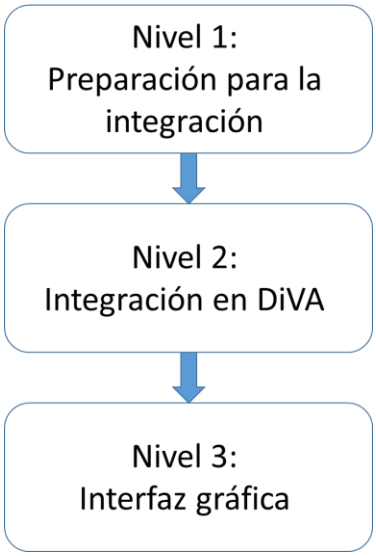


Figura 2-3: Niveles para la integración de un algoritmo en DiVA. Fuente: Propia.

En el nivel 1 se debe preparar el algoritmo para la integración posterior en DiVA. En este el algoritmo debe mostrar los resultados obtenidos después de procesar un video como entrada. Además, como requisitos adicionales, el algoritmo debe estar programado en C++ y debe de tener ciertos métodos.

En el segundo nivel el algoritmo debe de ser una clase que a su vez esté encapsulado dentro de una clase propia de DiVA encargada de, entre otras cosas, recibir imágenes de los servidores de esta plataforma. De cara al funcionamiento del algoritmo, lo único que cambia en este nivel es que hace uso de las imágenes que llegan desde los servidores en vez de utilizar una secuencia de video.

Finalmente, el tercer nivel aumenta la interactividad del algoritmo al implementarlo dentro de una interfaz gráfica. Este nivel no es estrictamente necesario ya que el algoritmo tiene la misma funcionalidad que en el nivel anterior, pero da más posibilidades de desarrollo.

Capítulo 3: GENERACIÓN DE LA IMAGEN PANORÁMICA DE FONDO

3.1 INTRODUCCIÓN Y APROXIMACIÓN

Como se ha comentado anteriormente, los segmentadores frente-fondo están diseñados para utilizarserse con imágenes captadas con cámaras fijas. El problema ocurre cuando el campo de visión de una cámara fija no es suficiente; para poder aumentarlo se utilizan cámaras PTZ que no son estáticas, son móviles pivotando desde su punto óptico. Según se describió en el Capítulo 2, si sobre una cámara móvil se aplica el citado algoritmo de segmentación, el resultado no es el esperado.

La solución propuesta para poder utilizar un algoritmo de segmentación con una cámara PTZ consta de dos fases (ver Figura 3-1). La primera fase es una fase de inicialización donde se genera una imagen panorámica de fondo, y la segunda es una fase de operación en la cual se va combinando la imagen panorámica junto a cada nueva imagen captada por la cámara PTZ para detectar con el segmentador frente-fondo el movimiento en la captura panorámica de la escena.

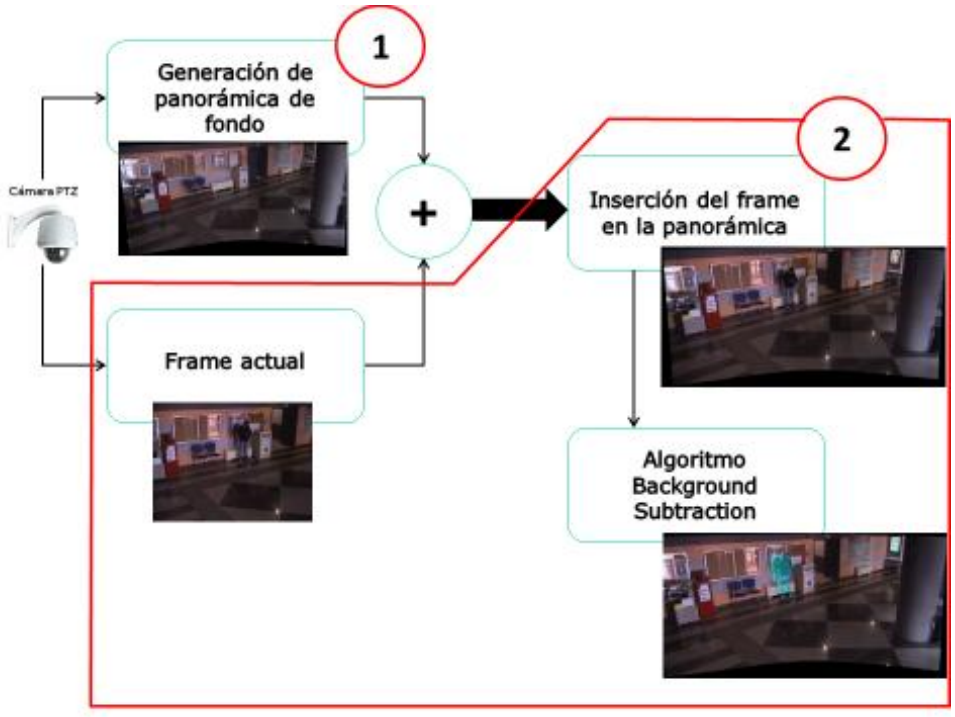


Figura 3-1: Esquema de la solución propuesta. Fuente: Propia

Para poder utilizar cámaras PTZ se propone modificar la entrada al algoritmo de *background subtraction*, en lugar de enviar directamente cada imagen captada por la cámara PTZ se utiliza una imagen panorámica generada a partir de múltiples imágenes captadas por la cámara PTZ y en esta imagen se inserta cada nueva imagen captada donde corresponda.

En este capítulo, se describirá detalladamente la primera fase del algoritmo que es la encargada de generar una imagen panorámica. La imagen panorámica generada para ser utilizada como base de las imágenes que se pasará al algoritmo de *background subtraction* debe de ser de la mayor calidad posible para que este algoritmo no detecte falsos positivos en su fase de operación.

Para calcular una imagen panorámica se sigue el diagrama de la Figura 3-2:

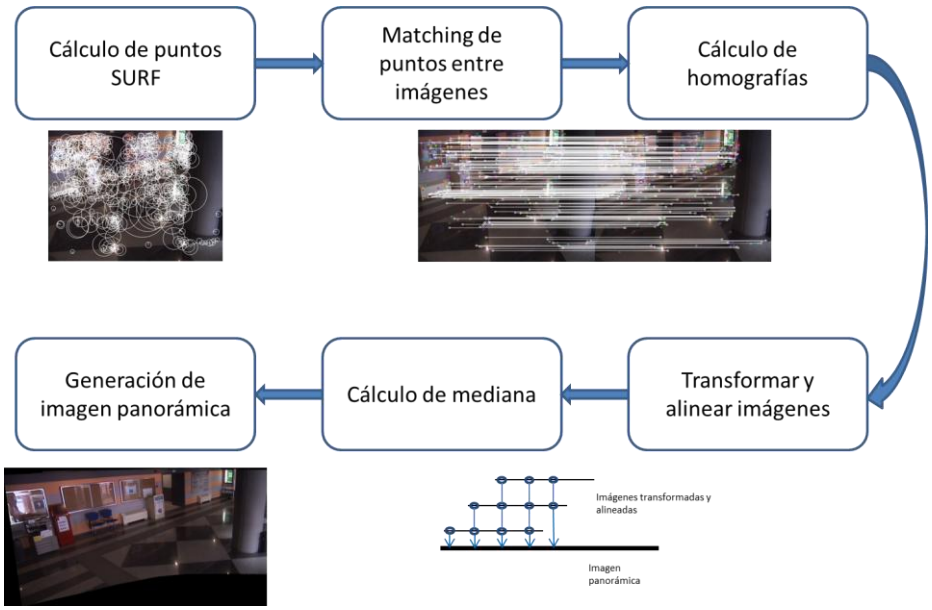


Figura 3-2: Esquema de generación de panorámica. Fuente: Propia

El primer paso para generar dicha imagen es la selección del punto de vista, el cual es la imagen en la que se base la imagen panorámica. A continuación se calculan los puntos SURF de cada imagen y se hace un *matching* entre imágenes consecutivas, el motivo de esto será explicado en la sección 3.3. Finalizado el *matching* de puntos se calcula la homografía y con ésta se transforma la imagen correspondiente. Para finalizar, una vez que se tienen todas las imágenes transformadas y alineadas se combinan de forma que el resultado final será una imagen panorámica.

A continuación se detallan los pasos descritos en el párrafo anterior.

3.2 AJUSTE DE LA CÁMARA PTZ

La aproximación empleada en este trabajo se basa en la generación de una imagen panorámica y para ello se deben tener en cuenta ciertos parámetros de la cámara. En este sentido, hay varios parámetros que son críticos: la velocidad a la que se mueve la cámara, el nivel de iluminación de la escena y el número de imágenes por segundo que se captan.

Antes de lanzar la fase de inicialización, que es la encargada de crear la imagen panorámica, se debe decidir la velocidad a la que la cámara se mueve y el número de imágenes que capta la cámara por segundo (*frame rate*).

En un principio, se desea generar la imagen panorámica en el menor tiempo posible, de forma que la variación de la escena durante el proceso sea mínima: que no aparezca una persona u objeto durante este proceso, ni haya cambios en la iluminación. El problema principal que tiene generar la imagen panorámica en el menor tiempo posible ocurre cuando la escena está poco iluminada. En este caso, al tener poca iluminación, las imágenes pueden resultar movidas y no alinearse perfectamente, por lo que el resultado será una imagen panorámica con zonas borrosas o poco definidas.

Para ello, en este trabajo se ha decidido establecer la velocidad más lenta posible de la cámara y capturar imágenes a un fps (*frames por segundo*), de forma que con la escena que se quiere trabajar, que cubre un área de 100º aproximadamente, cada par de imágenes se solapen un 90%.

3.3 SELECCIÓN DEL PUNTO DE VISTA

Normalmente, el primer paso para calcular una imagen panorámica es seleccionar el punto de vista desde el que va a realizarse la panorámica.

Un algoritmo que no operase con imágenes que van recibiéndose en tiempo real dispondría inicialmente de un número determinado de imágenes que cubriesen toda la escena y las cuales estuviesen solapadas parcialmente. El procedimiento habitual para minimizar distorsiones por interpolación sería seleccionar la imagen que se encontrase en el centro de la escena como el punto de vista escogido para la creación de la imagen panorámica.

El punto de vista es importante de cara a la calidad de la imagen panorámica generada; en una escena en interiores, como es el caso en este trabajo, con poca profundidad, los extremos de la imagen se irán distorsionando cuanto más amplia sea

la panorámica. Por ello, al seleccionar la imagen central como el punto de vista, se consigue reducir al máximo de lo posible la distorsión en los bordes de la imagen panorámica.

En este trabajo se desea trabajar sobre imágenes que van llegando en tiempo real y por ello no se puede seleccionar una imagen concreta como el punto de vista de la panorámica, sino se utiliza el primer *frame* que llega desde la cámara PTZ. El resultado es razonable siempre que la escena sobre la cual se quiere trabajar no dé lugar a una imagen panorámica excesivamente amplia; en este caso, incluso si se escoge un extremo de la escena como punto de vista de la imagen panorámica, la distorsión generada en el otro no es excesiva.

Si la escena sobre la que se quiere trabajar fuese muy amplia, se necesitaría un método con el cual no se comenzase a crear la imagen panorámica hasta que el *frame* inicial correspondiera al centro de la escena.

3.4 CÁLCULO Y COMPOSICIÓN DE HOMOGRAFÍAS

Una vez seleccionada la imagen correspondiente al punto de vista, esta imagen pasa a formar parte de la panorámica directamente, es decir, sin sufrir ninguna transformación. Sin embargo, las demás imágenes que capta la cámara PTZ que serán utilizadas para la generación de la imagen panorámica se deben ajustar o adaptar al punto de vista de la imagen inicial seleccionada.

Lo siguiente es ir calculando, para cada nueva imagen, la homografía que relaciona esta imagen o punto de vista con el escogido para la panorámica, aplicarla, y combinar la imagen transformada con la imagen panorámica (inicialmente una sola imagen).

Conforme la cámara se mueve, el cambio de punto de vista entre una nueva imagen y la panorámica va aumentando y el grado de solapamiento entre dicha imagen y la panorámica va disminuyendo. Por este motivo, la aplicación directa del algoritmo de cálculo de homografías detallado en la sección 2.2.2 arroja resultados que van degradándose.

La solución que se ha adoptado para minimizar los errores en el cálculo de homografías es la siguiente:

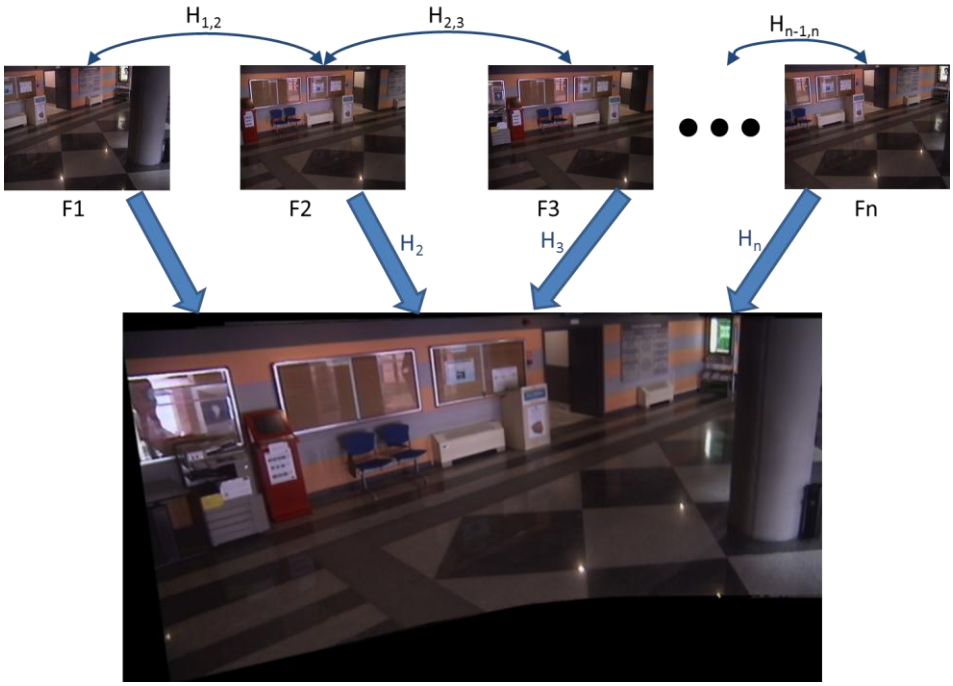


Figura 3-3: Diagrama de la composición de homografías. Fuente: Propia.

Sean *n frames*, recibidos consecutivamente de una cámara PTZ, con los cuales se quiere generar la imagen panorámica. El primer *frame, F1,* no se modifica y se inserta directamente en la imagen. Para el siguiente *frame*, F2, se calcula la homografía (H_2) que lo inserta en la imagen panorámica como la composición entre el homografía calculada anteriormente y la homografía entre el segundo *frame* y el primero ($H_{1,2}$), como el *frame* anterior no tiene homografía, se cumple que $H_2 = H_{1,2}$.

Para el tercer *frame*, F3, se calcula la homografía (H_3) que lo inserta en la imagen panorámica de la misma manera, como la composición entre la homografía anterior (H_2) y la homografía entre el tercer *frame* y el segundo ($H_{2,3}$), de manera que se cumple que $H_3 = H_2 * H_{2,3}$, así sucesivamente hasta adaptar el *frame* Fn con la homografía $H_n = H_{n-1} * H_{n-1,n}$.

De esta manera, siguiendo el esquema de la Figura 3-3, se transforman y se alinean todas las imágenes que compondrán la imagen panorámica.

Durante el proceso del cálculo de homografías entre imágenes consecutivas descrito anteriormente, ha habido problemas que han tenido que ser solucionados para el funcionamiento correcto del algoritmo. Estos problemas pueden ser muy graves ya que, al componer homografías, en el momento que una se calcula de manera defectuosa su error se propaga para todas las siguientes y por lo tanto el resultado final sería una panorámica de baja calidad.

El primer problema abordado surgió en el cálculo de los puntos característicos en las imágenes cuya homografía se desea obtener. Al calcular los puntos SURF estos

deben estar dispersos por toda la imagen; esto se debe a que en caso de tener una agrupación de puntos en una única zona de la imagen puede resultar en que la homografía calculada sea poco precisa, lo que resultaría en una transformación razonablemente correcta para esa zona de la imagen pero no necesariamente para toda la imagen.

Para solucionar el problema comentado en el párrafo anterior, se probaron dos técnicas distintas. La primera consiste sencillamente en bajar el umbral de detección de puntos SURF (cuanto más bajo más puntos característicos se detectan), de manera que se obtienen miles de puntos por toda la imagen, lo cual hace poco probable que todos cubran una única zona.

La segunda técnica consiste en calcular los puntos SURF de la imagen con un cierto umbral, luego dividir la imagen en 3x3 bloques de igual tamaño y comprobar si todos los bloques contienen puntos característicos. Si no tienen puntos característicos todos los bloques se baja el umbral y se vuelve a comprobar y así sucesivamente hasta que todos los bloques contengan al menos un punto característico. Debido a que hay bloques donde no es necesario bajar el umbral para que tengan puntos, cuando se baje el umbral y todos los bloques tengan puntos característicos, estos bloques pueden tener cientos de puntos, pero sólo se mantienen los cien primeros puntos donde el umbral no era tan bajo. De esta manera se consigue tener puntos característicos distribuidos por toda la imagen con un umbral efectivo diferente para cada bloque de la imagen.

De los dos métodos descritos se decidió utilizar finalmente el primero ya que al obtener miles de puntos se puede asumir que hay puntos distribuidos por toda la imagen y porque con el segundo método el tiempo que se tardaba en obtener los puntos era demasiado elevado para una aplicación de tiempo real.

El segundo problema surge en el momento en que se hace el *matching* de puntos entre dos *frames* consecutivos a la hora de a la hora de obtener la homografía que los relaciona. Después de hacer el *matching* solo se aceptan los emparejamientos con distancia *x* veces menor o igual que la distancia mínima calculada en el *matching*.

Si la distancia es demasiado pequeña la homografía se obtendrá a partir de puntos muy fiables pero muy pocos, y por lo tanto la distribución de puntos característicos por toda la imagen, calculada anteriormente, puede verse afectada; si la distancia, por el contrario, es demasiado grande se aceptarán muchos emparejamientos en los cuales los puntos pueden ser muy parecidos pero no el mismo de forma que la homografía puede ser defectuosa.

La solución que se ha propuesto a este problema es meter en un bucle el paso en el cual se decide si un emparejamiento es bueno o malo. En dicho bucle se empieza con una restricción alta sobre la distancia mínima que se debe de cumplir y se calcula cuantos emparejamientos se consideran buenos; si no son suficientes se relaja un poco la restricción y se vuelve a contar cuantos emparejamientos buenos hay, y así se sigue

hasta que llega un momento en el que hay suficientes puntos buenos para calcular la homografía. Para este trabajo se ha considerado que con cuarenta emparejamientos se obtiene un equilibrio entre número de puntos característicos en la imagen y número de emparejamientos fiables.

3.5 *MERGING* O CONCATENACIÓN DE IMÁGENES EN LA PANORÁMICA

A continuación, una vez que se han calculado las homografías correspondientes y se han transformado y alineado las imágenes que compondrán la imagen panorámica, se generarán zonas de la panorámica en las que coincidan o se solapen dos o más imágenes (ver Figura 3-4). Para decidir qué valor se da en estos casos al píxel resultante en la panorámica final se plantearon dos opciones distintas.

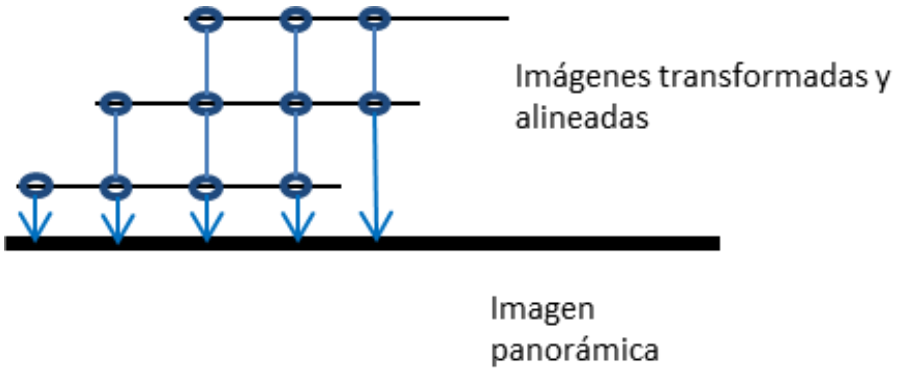


Figura 3-4: Esquema del cálculo de la mediana. Fuente: Propia.

La primera consiste en calcular la media de los píxeles que se solapan en una determinada posición. Esta solución es muy rápida de calcular. El problema observado es que la media hace que el resultado final sea una versión suavizada o promediada de cómo es la escena en cada imagen y píxeles de la imagen panorámica final pueden recibir un valor que no se ha obtenido en ninguna de las imágenes en ese punto.

La segunda consiste en calcular la mediana de los píxeles que se solapan en una determinada posición. Para ello la mediana ordena el vector con todos los valores de los píxeles en ese punto y selecciona el que se encuentra en medio y por lo tanto el valor que toma la panorámica en cada píxel es uno que se ha obtenido en alguna imagen, el que coincide con el centro de su distribución.

Es por ese motivo por el cual se descartó utilizar la media, dando preferencia a generar una imagen con la mayor precisión posible sacrificando velocidad.

Un problema importante a corregir, que ocurre aquí, es el resultante de las condiciones variables de la iluminación de la captura. La cámara PTZ utilizada en este trabajo tiene un control automático de ganancia lo que resulta en que una zona de la escena puede tener diferentes niveles de luminancia dependiendo del punto de vista desde el que se capture. El resultado de no corregir este efecto es la aparición de bandas en la panorámica, bandas que el segmentador frente-fondo puede marcar como frente a pesar de ser fondo.

La solución propuesta para resolver este efecto es obtener una máscara que indica qué zonas entre cada nueva imagen y la imagen panorámica están solapadas; luego se calcula la luminancia media de ambas imágenes en esta zona y se suma la mitad de la diferencia de luminancia media entre ambas imágenes a la imagen que se quiere añadir a la panorámica. Aunque se trata de una aproximación muy simple, de esta manera, la imagen a insertar en la panorámica tendrá una luminancia más parecida a la imagen panorámica y por lo tanto se mitigará el posible cambio de luminancia.

3.6 GENERACIÓN DE PANORÁMICAS EN PRESENCIA DE OBJETOS EN MOVIMIENTO

Según se ha comentado en la introducción de este capítulo, el objetivo de generar una panorámica de fondo es utilizarla como base para generar las imágenes que se va a utilizar para alimentar el algoritmo de segmentación frente-fondo. Por lo tanto, esta panorámica debe estar libre de objetos de frente. Dado que al generar la imagen panorámica no se puede garantizar que no haya objetos de frente o en movimiento, es necesario desarrollar una técnica para poder generar la panorámica en estas circunstancias.

Una vez obtenidos puntos característicos dispersos por toda la imagen pero antes de hacer el *matching* de puntos, puede ocurrir que haya puntos situados en zonas inestables de la imagen. Estas zonas inestables pueden ser zonas donde ha pasado un objeto mientras que se genera la panorámica. Si en un *frame* captado por la cámara PTZ hay una persona se obtendrán puntos SURF en la persona y su contorno; si en el siguiente *frame* está persona se mueve ligeramente, al hacer el *matching* de los puntos entre el par de imágenes, la homografía resultante será ligeramente defectuosa.

Para poder realizar la generación de la panorámica sin que el movimiento de los objetos deteriore el resultado final, se calcula la media y la varianza de cada píxel de la imagen panorámica cada vez que se actualice con una nueva imagen, luego se crea una máscara donde ésta vale uno donde la varianza es menor que cinco y cero donde la varianza es mayor. Finalmente, todos los puntos característicos calculados que estén donde la máscara es cero son descartados y de esta manera no se utilizan puntos característicos a no ser que sea una zona “estable” de la imagen panorámica.

Finalmente, en la Figura 3-5 se muestran un posible resultado de la imagen panorámica para la fase de inicialización del algoritmo.

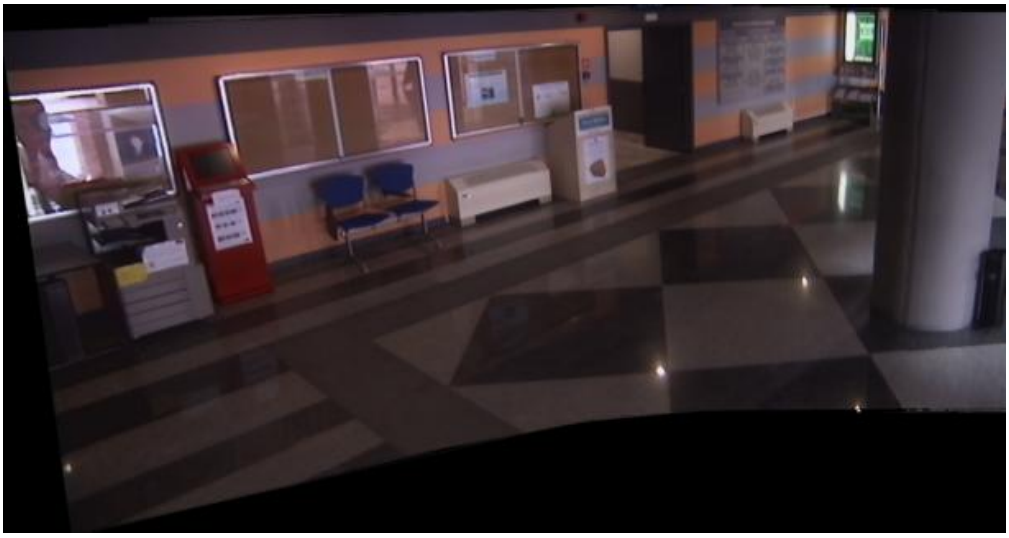


Figura 3-5: Imagen panorámica final. Fuente: Propia.

Capítulo 4: INTEGRACIÓN CON ALGORITMOS DE *BACKGROUND SUBTRACTION*

4.1 INTRODUCCIÓN

Como se ha dicho en el capítulo 1 de esta memoria, el objetivo principal de este trabajo es aumentar el campo de visión de un algoritmo de *background subtraction* utilizando una cámara PTZ en movimiento en lugar de una cámara fija.

Según se indicó en el capítulo anterior, la solución propuesta en este TFG consta de dos fases (ver Figura 3-1): una fase de inicialización, en la que se genera una imagen panorámica del campo de observación abarcado por la cámara PTZ; y una fase de operación, en la que cada nueva imagen de la cámara se *inserta* en la panorámica y se utiliza la panorámica resultante como entrada de un algoritmo de *background subtraction*, haciendo de este modo operar a este algoritmo con una imagen mucho mayor que la procedente de la cámara.

En el capítulo anterior se ha explicado cómo se obtiene una imagen panorámica a partir de las imágenes captadas por la cámara PTZ y cómo se han solventado los distintos problemas que han surgido, (fase 1 de la Figura 3-1). En este capítulo se describe la solución propuesta para poder utilizar la imagen panorámica generada anteriormente como base para integrar en ella las nuevas imágenes captadas por la cámara y alimentar un algoritmo de *background subtraction,* (fase 2 de la Figura 3-1).

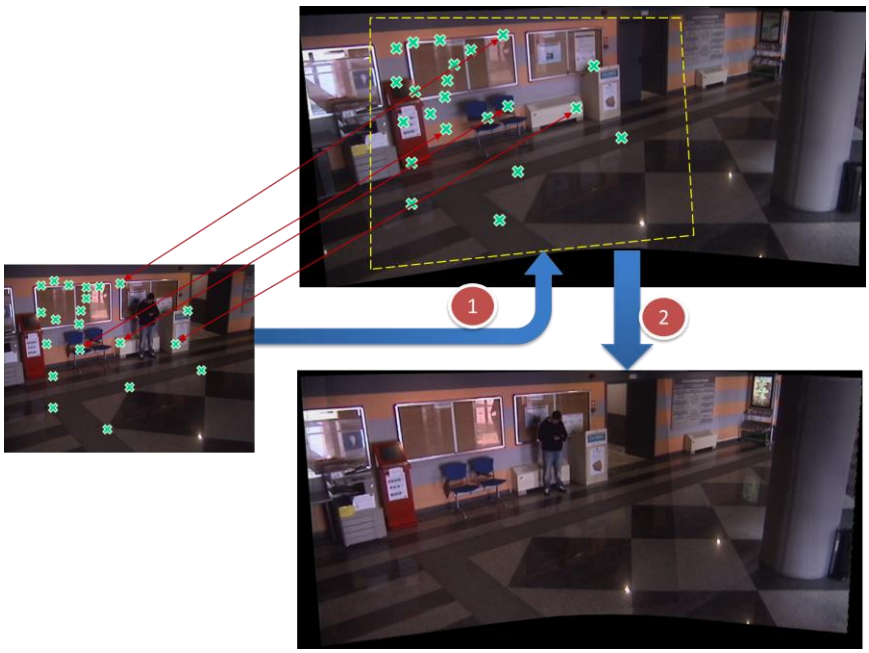


Figura 4-1: Esquema de la solución propuesta. Fuente: Propia

Para ello se sigue el esquema de la Figura 4-1, en el cual se busca la zona de la panorámica a la que pertenece la imagen que llega desde la cámara y a continuación se inserta ahí.

4.2 INICIALIZACIÓN DEL ALGORITMO DE *BACKGROUND SUBTRACTION*

Como se ha comentado anteriormente, en este trabajo se quiere aumentar la funcionalidad de un algoritmo de *background subtraction* cualquiera utilizando una cámara PTZ. Para ello se ha decidido hacer uso de un segmentador frente-fondo que utiliza el método gamma debido a su rapidez y resultados, todo ello detallado en [19].

Como la mayoría de los algoritmos de *background subtraction*, el que se ha escogido necesita inicializar el modelo de fondo para poder detectar el frente en el *frame* que se recibe desde la cámara. Para ello es conveniente lanzar inicialmente el algoritmo con una imagen panorámica libre de objetos en movimiento, que es la imagen resultante de los procesos descritos en el Cap. 3.

Es fundamental que esta panorámica sea correcta y sin fallos, ni objetos que no pertenezcan al fondo ya que es la imagen modelo de fondo que utilizará el segmentador frente-fondo y cualquier defecto en esta imagen podría resultar en que el segmentador lo considere frente.

4.3 INSERCIÓN DEL *FRAME* ACTUAL EN LA PANORÁMICA DE FONDO

Cada nueva imagen que llega de la cámara PTZ se inserta o combina con la panorámica de fondo, y la imagen resultante se usa como entrada para el algoritmo de *background subtraction*. Dado que la panorámica de fondo corresponde a un instante temporal previo y cada vez más anterior al de la imagen actual, si varían de algún modo las condiciones de iluminación de la escena, la inserción directa de la nueva imagen en el fondo panorámico sería visiblemente apreciable, incluso en ausencia de objetos en movimiento, lo cual generaría falsas alarmas o frente inexistente en el algoritmo de *Background subtraction*. Este efecto también podría producirse por cambios de iluminación debidos al control automático de ganancia de la cámara.

Para evitarlo, se ha propuesto mantener un modelo de la variación de luminancia de cada píxel de la panorámica de fondo, de modo que al combinar la

imagen actual con la citada panorámica, sólo se inserten los píxeles que se considere que han sufrido un cambio notable.

Más en detalle, a la hora de insertar el nuevo *frame* procedente de la cámara se sigue un procedimiento muy similar al utilizado a la hora de generar la panorámica. Primero se tiene que saber dónde en la imagen panorámica debe *colocarse* el *frame* actual y luego se debe hacer una decisión sobre que píxeles se deben insertar.

Para conocer donde situar el *frame* nuevo en la imagen se calcula en una primera aproximación la homografía que sitúa, de manera *grosera*, este nuevo *frame* dentro de la imagen panorámica. Para ello se calculan los puntos SURF en toda la imagen al igual que en toda la imagen panorámica, se hace el *matching* de estos puntos y con ello se calcula la homografía. Se indica que esta homografía puede ser *grosera* porque si en la escena capturada por el frame actual hay un objeto de frente, es muy probable que en el contorno de dicho objeto se seleccionen puntos SURF que no van a encontrar correspondencia en la panorámica de fondo.

Para obtener una homografía más precisa utilizamos la media y varianza de cada píxel de la imagen panorámica, calculados anteriormente, con el objetivo de descartar puntos SURF en zonas del *frame* nuevo donde sospechamos que hay frente. Para ello, se transforma la imagen con la estimación menos precisa de la homografía y se genera una máscara donde ésta vale uno donde el valor absoluto entre la resta del valor del píxel del *frame* nuevo, p(i,j), y la media en ese píxel, de la imagen panorámica, μ(i,j), es menor o igual que X veces (se ha tomado X=5) la deviación típica, sigma(i,j), de los valores que toma ese píxel (ver Ecuación 3); en caso contrario la máscara en ese píxel vale cero. Una vez calculada esta máscara se descartan todos los puntos característicos del *frame* nuevo donde la máscara vale cero, ya que son estas zonas donde sospechamos que hay frente, y se vuelve a calcular la homografía.

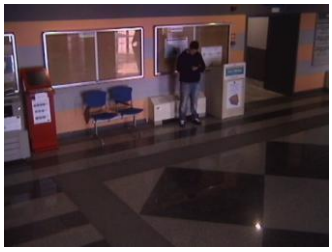
$$|p\ (i,j) - \mu\ (i,j)| \leq X * \sigma\ (i,j) \qquad (3)$$

Una vez que sabemos dónde debe estar situado el *frame* nuevo en la imagen panorámica se podría insertar directamente reemplazando los valores de la imagen panorámica por los del nuevo *frame*. Sin embargo, se ha optado por no insertar la imagen directamente e introducimos dos pequeñas modificaciones con el objetivo de mejorar la posterior detección de frente del segmentador.

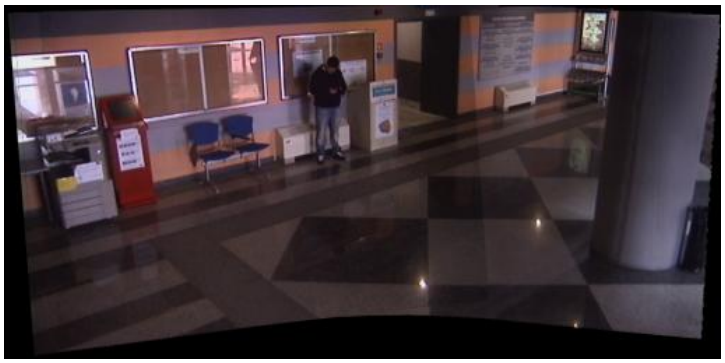
La primera modificación es ajustar la luminancia media del nuevo *frame* de manera muy similar a como se hacía cuando se generaba la imagen panorámica. Para ello calculamos la luminancia media del *frame* nuevo y la luminancia media de la zona de la imagen panorámica donde se insertará dicho *frame*. Una vez calculado este valor, se suma la mitad de la diferencia al nuevo *frame* con la intención de que el

segmentador no marque como frente un cambio de iluminación debido principalmente al efecto del control automático de ganancia.

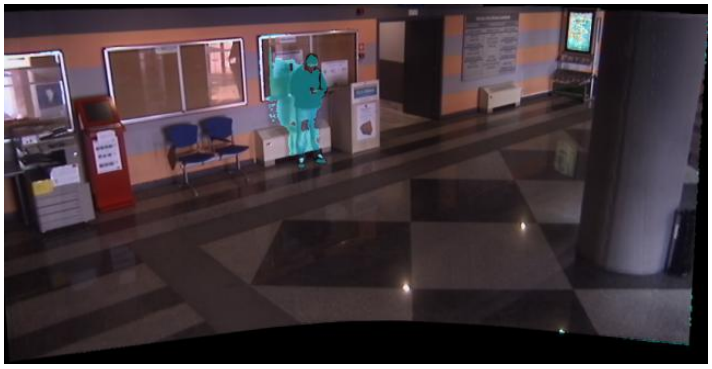
La segunda modificación está también pensada para ayudar al segmentador a decidir que es frente y que es fondo. Lo que se hace es insertar en la imagen panorámica todos los píxeles cuyo valor supera un umbral pero se mantienen aquellos que no. Para ello se crea una máscara donde ésta toma el valor de cero donde se cumple la Ecuación 3, y uno en el resto.



a)



b)



c)

Figura 4-2: (a) Frame captado con frente de la cámara PTZ, (b) Frame insertado en la imagen panorámica, (c) Resultado del segmentador gamma. Fuente: Propia.

Al crear la máscara descrita en el párrafo anterior, el valor de X es por defecto igual a tres. Si este parámetro se aumenta se considerará que hay más fondo y menos píxeles serán remplazados; en el caso contrario, si se baja el umbral, se considera que hay más frente y más píxeles serán reemplazados. Una vez obtenida esta máscara binaria, se le aplica la erosión morfológica con un elemento estructurante cuadrado de 3x3 para eliminar espurios.

Finalizado el proceso de calcular la máscara, se reemplazan en la imagen panorámica los píxeles donde la máscara es igual a uno y es esta imagen panorámica la que se utiliza como entrada al algoritmo de *background subtraction* el cual devuelve una máscara indicando donde hay frente. Para visualizar mejor está máscara de frente se le suma a la imagen panorámica de fondo y así se puede ver fácilmente las diferencias entre la imagen panorámica creada anteriormente y el *frame* que llega desde la cámara (ver Figura 4-2).

4.4 ACTUALIZACIÓN DE LA PANORÁMICA DE FONDO

El proceso anterior se realiza para cada *frame* que se recibe procedente de la cámara PTZ; pero, según se ha comentado, el modelo o panorámica de fondo en la que se insertan estas imágenes se debe de actualizar periódicamente para así ajustarse a una posible evolución temporal de la iluminación o a cambios de los objetos que componían el fondo.

Para realizar dicha actualización de la panorámica de fondo se utiliza un proceso muy similar al utilizado anteriormente, previo al segmentador frente-fondo, para decidir que es fondo y que es frente. Primero de todo se hace una aproximación de la homografía que transforma la imagen procedente de la cámara para obtener su localización dentro de la imagen panorámica, luego se obtiene una máscara que indica dónde se puede asegurar que hay frente y se vuelve a calcular una homografía más precisa eliminando los puntos característicos que se encuentran en una de esas zonas.

Una vez colocado, de la forma más precisa posible, el *frame* procedente de la cámara se calcula nuevamente una máscara que esta vez indica qué puede ser frente. Para ello, de la misma manera que antes, a los píxeles de la máscara se les asigna el valor de cero donde se cumple la Ecuación 3 y uno en el resto.

Anteriormente, para eliminar los puntos característicos que se encontrasen en el frente el valor del parámetro X era igual a cinco; pero eso no garantizaba que todo lo que estuviese por debajo fuese fondo. En este caso la condición podría ser que X fuese igual a uno, de modo que todos los píxeles que cumplen la condición fueran fondo con gran probabilidad; pero debido a que puede haber cambios de luminosidad en la

escena, estos cambios se quieren tener en cuenta para así mejorar la actualización del fondo y por ello la condición se decidió que fuera más laxa, haciendo que X valga tres.

Con esta nueva máscara se puede actualizar la imagen panorámica de fondo. Para ello se crea una nueva imagen y se le asignan el valor a los píxeles dependiendo de la máscara: donde vale uno se considera frente y por lo tanto a la nueva imagen se le asignan el valor de los píxeles de la panorámica, donde la máscara vale cero se considera que puede ser fondo y por lo tanto a la nueva imagen se le asigna el valor de los píxeles de la imagen captada por la cámara PTZ.

Una vez obtenida esta imagen se inserta en el vector de imágenes, con el cual se calcula la imagen, panorámica, se elimina la imagen más antigua de este vector y se vuelve a calcular la imagen panorámica, su media y su varianza. A pesar de tener una condición más laxa que puede resultar en que zonas de frente se utilicen para actualizar la imagen panorámica no debería de haber problemas gracias a que el valor de los píxeles de la imagen panorámica final se calculan como la mediana de las imágenes transformadas y alineadas y mientras que un objeto que no es fondo se quede quieto mucho tiempo, éste no saldrá en la imagen panorámica.

Este proceso de actualización de la imagen panorámica no se recomienda hacer por cada *frame* que capta la cámara PTZ debido a que el coste computacional de este proceso hace que el algoritmo se ralentice considerablemente. El periodo con el que se debe actualizar el fondo se mide en *frames* y es una variable que se debe ajustar teniendo en cuenta la variabilidad de la escena sobre la que se quiere trabajar y el número de *frames* por segundo (*frame rate*) que capta la cámara.

Por ejemplo, si se tiene una escena en la cual el la iluminación del fondo varia lentamente y la cámara capta imágenes a 8 *frames* por segundo se ha observado que es recomendable actualizar la panorámica de fondo en torno a cada 100 *frames;* si por el contrario el fondo cambia rápidamente y la cámara capta 16 *frames* por segundo es recomendable que se actualice el fondo cada 30-50 *frames*.

Capítulo 5: PRUEBAS

Una vez terminado el desarrollo del algoritmo se ha procedido a hacer pruebas sobre el mismo midiendo la calidad de la imagen panorámica y el tiempo de procesado necesario para obtenerla. Todas estas pruebas no pueden ser cuantitativas debido a que no se dispone de un *ground truth* sobre la imagen panorámica: por lo tanto la medida de calidad de dicha imagen es subjetiva. Por el mismo motivo, las detecciones que hace el segmentador frente-fondo son subjetivas, es decir, ejemplos de aplicación en los que es posible observar la calidad de su comportamiento.

Las medidas que sí son cuantitativas son las mediciones del tiempo de procesado que tarda cada parte del algoritmo, dato de especial relevancia en un sistema de tiempo real. Para estas medidas el equipo que se ha utilizado tiene un procesador Intel® Core™ i5-3330 con una CPU a 3 GHz y 8GB de RAM. El sistema operativo es Windows 8 de 64 bits.

El algoritmo primero ha sido implementado en MatLab 2012b y posteriormente en C++ con OpenCV versión 2.4.6, con la cual se han hecho todas las medidas de tiempo de ejecución.

5.1 EVALUACIÓN DE LAS PANORÁMICAS

La generación de la imagen panorámica se pensó en un principio con el objetivo de que fuese en tiempo real pero con resultados de alta calidad, ya que ésta se ha comprobado que es crítica para el buen funcionamiento del algoritmo completo. A lo largo del desarrollo de este trabajo se ha visto la necesidad de incorporar mejoras para aumentar la calidad de la imagen panorámica generada, lo que ha resultado que esta fase del algoritmo tenga un tiempo de procesado demasiado alto para considerarlo tiempo real. En cualquier caso, ello no afecta a la funcionalidad del sistema, ya que la generación de esta imagen se realiza en una etapa de inicialización previa a la etapa de operación.

La Tabla 5-1 muestra los tiempos de ejecución de esta fase del algoritmo para 10 pruebas con la cámara grabando a un fps (*frames per second*) y completando dos barridos completos de la escena para generarla.

Número de prueba	Tiempo medio en Calcular Homografías (segundos por *frame*)	Tiempo total en calcular mediana (segundos)	Tiempo medio en calcular media y varianza (segundos por *frame*)	Numero de *frames* en panorámica	Tiempo total en generar la panorámica (segundos)
1	0.2189	1.1808	0.3839	27	25.1320
2	0.2228	0.9377	0.3103	22	18.8299
3	0.2092	0.9724	0.3194	23	19.3004
4	0.2135	0.8844	0.2977	21	17.1372
5	0.2266	1.1694	0.3708	26	23.4406
6	0.2144	1.0324	0.3388	24	20.0054
7	0.2197	0.8059	0.2797	19	15.1681
8	0.2195	0.8889	0.2911	21	17.3388
9	0.2213	0.9949	0.3183	23	19.5845
10	0.2225	1.0519	0.3358	24	21.3295
Tiempo medio (segundos)	**0.21884**	**0.9919**	**0.3246**	**23**	**19.7266**

Tabla 5-1: Evaluación de tiempo de procesado de la primera fase a 1 fps.

De la citada tabla concluimos que la tarea que más tiempo tarda en completarse es el cálculo de la mediana. Este es tan elevado que provoca que no se puedan ver los resultados parciales si se quiere trabajar a 1 fps.

Por otro lado podemos comprobar que el tiempo medio en procesar un *frame* para generar la imagen panorámica es de 0,8577 segundos, lo que implica que se pueden procesar 1,17 fps.

Las imágenes panorámicas generadas con la cámara captando imágenes a un fps, completando un barrido completo de la escena desde distintos puntos de visión son las siguientes:

a)

b)

c)

Figura 5-1: Resultados obtenidos teniendo el punto de vista en (a) el extremo derecho, (b) el centro, (c) extremo izquierdo, con la cámara captando a 1 fps y un barrido completo de la escena. Fuente: Propia.

Como se puede ver en las imágenes, la distorsión en los extremos de la imagen panorámica son menores cuando se empieza desde el centro de la escena, pero aun así, debido a que la escena no es muy amplia, los extremos no se deforman excesivamente y el resultado final es una panorámica de una calidad buena.

Las panorámicas de la Figura 5-1 están generadas solamente a partir de un barrido completo de la escena. Las imágenes panorámicas de la Figura 5-2 han sido generadas con los mismos parámetros, excepto el número de barridos que han sido dos.

a)

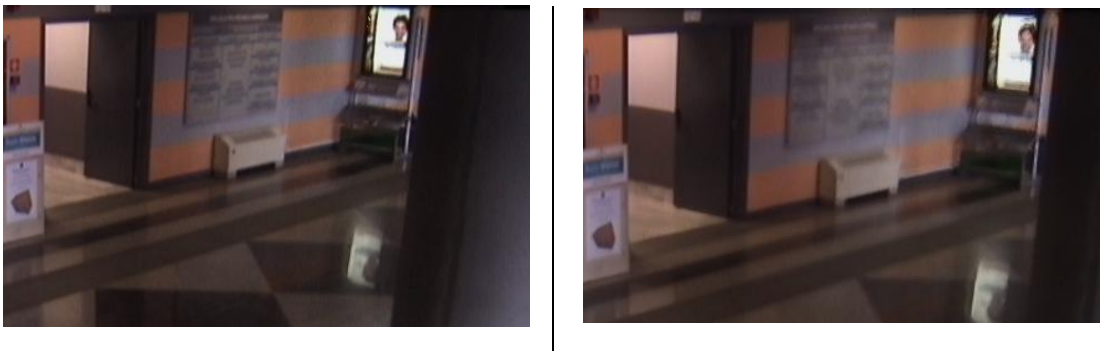
b)

c)

Figura 5-2: Resultados obtenidos teniendo el punto de vista en (a) el extremo derecho, (b) el centro, (c) extremo izquierdo, con la cámara captando a 1 fps y dos barridos completos de la escena. Fuente: Propia.

Al igual que ocurría anteriormente, las imágenes se distorsionan menos cuando el punto de vista de la imagen panorámica está en el centro de la escena, pero nunca llega a ser excesiva la distorsión de los extremos cuando no se comienza desde el centro.

También podemos comparar los resultados obtenidos entre las imágenes panorámicas obtenidas generadas a partir de uno o dos barridos completos. Al generar dichas imágenes con dos barridos completos de la escena se puede observar que hay zonas que no están perfectamente definidas, y comparadas con la misma imagen generada a partir de un barrido éstas últimas son de mejor calidad (ver Figura 5-3).



a) b)

Figura 5-3: Comparativa entre imagen generada con (a) un barrido y (b) dos barridos. Fuente: Propia.

Esta falta de calidad en la imagen generada a partir de dos barridos es debido a que el tiempo que tarda en generarse la imagen es mayor y por lo tanto hay una

mayor probabilidad de que pase una persona durante el proceso y por lo tanto, pequeños fallos a la hora de calcular las homografías se propagan y entonces las imágenes no se alinean perfectamente y resultan en estos pequeños defectos. Es por ello que se recomienda generar la imagen panorámica con aproximadamente 20 imágenes, en un corto plazo de tiempo.

5.2 EVALUACIÓN DEL ALGORITMO COMPLETO

Una vez evaluada la primera fase del algoritmo (generación de la imagen panorámica), se procede a evaluar la segunda fase. Esta segunda fase es la encargada de insertar el *frame* en la imagen panorámica y utilizarlo como entrada para el algoritmo de *background subtraction*.

Como se ha comentado anteriormente, la evaluación de esta fase es en parte subjetiva y en parte objetiva. La evaluación subjetiva es la que se hace de los resultados que se obtienen a la salida del segmentador frente-fondo debido a que no se dispone de *ground truths* con los que comparar los resultados. La evaluación cuantitativa es la medición del tiempo que tarda en ejecutarse el algoritmo y las diversas fases que lo componen.

La Tabla 5-3 muestra los tiempos de ejecución de esta fase del algoritmo para 10 pruebas con la cámara grabando a 8 fps (*frames per second*) y con la imagen panorámica generada a partir de 1 barrido de la escena (en cada una el punto de vista puede ser distinto).

De los datos obtenidos de la Tabla 5-3, se puede calcular que esta segunda fase es capaz de procesar 1,12 *frames* por segundo con todas las mejoras incorporadas.

Debido a que el tiempo de procesado de un *frame* en esta fase es alto, se procede a hacer una evaluación de tiempo de las mejoras insertadas en esta fase que son las causantes de que aumente el tiempo de procesado de cada *frame*.

En la Tabla 5-4 se muestran los tiempos de ejecución para 5 pruebas con la cámara grabando a 8 fps (*frames per second*) y con la imagen panorámica generada a partir de 1 barrido de la escena (en cada una el punto de vista puede ser distinto), pero sin calcular la homografía más precisa en un caso y sin incluir mejoras en el otro.

A partir de la Tabla 5-4 se demuestra que las mejoras insertadas para que el algoritmo completo funcione mejor, incrementan el tiempo hasta casi el doble, que principalmente se debe al proceso que calcula de forma más precisa la homografía.

Número de prueba	Tiempo medio insertar el *frame* (segundos por *frame*)	Tiempo medio en compensar iluminación (segundos por *frame*)	Tiempo de proceso del BGS (segundos por *frame*)	Tiempo medio en actualizar el fondo (segundos)	Tiempo total en procesar un *frame* (sin incluir actualización)
1	0.8938	0.0147	0.0235	1.5567	0.9216
2	0.8984	0.0137	0.0213	1.3682	0.8607
3	0.9743	0.0137	0.0217	1.3946	1.0006
4	0.7638	0.0129	0.0199	1.3149	0.7882
5	0.8724	0.0122	0.0189	1.5208	0.8962
6	0.9257	0.0139	0.0204	1.4822	0.9462
7	0.7994	0.0142	0.0196	1.3648	0.8310
8	0.8960	0.0128	0.0211	1.3340	0.8841
9	0.8747	0.0133	0.0209	1.4447	0.9021
10	0.9015	0.0137	0.0191	1.4206	0.8914
Tiempo medio (segundos)	**0.8800**	**0.0135**	**0.0206**	**1.4202**	**0.8922**

Tabla 5-2: Evaluación de tiempo de procesado de la segunda fase a 8 fps, con todas las mejoras.

	Sin calcular la homografía más precisa		Sin incluir ninguna mejora	
Número de prueba	Tiempo medio insertar el *frame* (segundos por *frame*)	Tiempo total en procesar un *frame* (sin incluir actualización)	Tiempo medio insertar el *frame* (segundos por *frame*)	Tiempo total en procesar un *frame* (sin incluir actualización)
1	0.4998	0.5251	0.4573	0.4805
2	0.4884	0.5135	0.3858	0.4096
3	0.4401	0.4640	0.3695	0.3941
4	0.4301	0.4539	0.3931	0.4186
5	0.4137	0.4384	0.4906	0.5195
Tiempo medio (segundos)	**0.4544**	**0.4789**	**0.4193**	**0.4445**

Tabla 5-3: Evaluación de tiempo de procesado de la segunda fase a 8 fps, sin todas las mejoras.

El algoritmo podría funcionar sin las mejoras insertadas en la segunda fase, pero obteniendo una cantidad de errores más elevada a la salida del segmentador frente-fondo.

A continuación se muestran unos ejemplos de la salida del segmentador utilizando o no las mejoras.

a)

b)

Figura 5-4: Resultados obtenidos teniendo a la salida del segmentador con (a) todas las mejoras, (b) sin mejoras. Fuente: Propia.

Como podemos observar en la Figura 5-3, al no incluir mejoras en la fase de operación del algoritmo, no se calcula la homografía más precisa y por lo tanto la mayoría de los contornos pueden no estar ajustados correctamente; por ello el segmentador marcará esas zonas como si hubiese movimiento. Este error no ocurre constantemente, pero ocurre con una cierta frecuencia que hace preferible que el algoritmo tarde más tiempo en procesar para evitar falsos positivos.

Capítulo 6: CONCLUSIONES Y TRABAJO FUTURO

6.1 CONCLUSIONES

En este trabajo se ha buscado incrementar el campo de visión de un algoritmo de detección de intrusos que utiliza un segmentador frente-fondo haciendo uso de una cámara PTZ, y todo ello en tiempo real. Una vez completado el desarrollo y la evaluación del algoritmo se puede llegar a varias conclusiones.

La solución propuesta es válida: con este algoritmo se puede detectar intrusos en una escena más amplia que el campo de visión de una cámara fija, pero tiene algunas limitaciones.

Una limitación al funcionamiento del algoritmo sucede al no poder ver lo que ocurre en toda la escena todo el tiempo. Al tener la cámara pivotando, si la velocidad de su movimiento es demasiado baja, podría darse el caso de que la cámara este apuntando a un extremo de la escena y un intruso pase por el otro extremo rápidamente de forma que no sería detectado.

La siguiente limitación es que las dos fases del algoritmo, la de inicialización y la de operación, se pueden ejecutar en tiempo real pero siempre que el *frame rate* sea bajo. Por este motivo, el algoritmo puede no detectar movimiento si un objeto o una persona pasa muy rápido (en el tiempo entre *frames*).

Para la mejora futura de la fase de inicialización, se recomienda crear una panorámica a partir de muchas más imágenes que las utilizadas en este trabajo, empezando desde el centro para evitar distorsiones en los bordes, aunque no sea posible realizarlo en tiempo real. El motivo principal es porque la imagen panorámica es fundamental para garantizar el funcionamiento correcto de la siguiente fase del algoritmo.

La segunda conclusión que se puede sacar de este trabajo es que la segunda fase de este algoritmo, la fase de operación, funciona y hace lo que se espera de ella pero el tiempo de procesado por cada *frame* puede resultar demasiado alto para ciertos escenarios. Para que pueda funcionar más rápido es necesario que la inserción de la imagen sea lo más simple posible.

Para poder realizar esto ayudaría tener una imagen panorámica con la mayor calidad posible de manera que no sea necesario calcular la homografía dos veces ya que este proceso es el que más tiempo requiere.

En conclusión, la solución propuesta es válida, el algoritmo es capaz de detectar intrusos a pesar de sus posibles limitaciones. Es posible solucionar dichas limitaciones si el código se optimiza y se trabaja sobre un equipo más potente.

6.2 TRABAJO FUTURO

Debido a la duración limitada de este trabajo, hay aspectos que no se ha podido llevar a cabo y que se pueden hacer en un futuro para mejorar el funcionamiento general del algoritmo.

En primer lugar hay métodos que no se han probado que podrían resultar en una mejora significativa de la imagen panorámica de fondo.

Para ello una idea que se propone es utilizar el zoom de la cámara PTZ, que no ha sido utilizado a lo largo de este trabajo, para aumentar la calidad de la imagen panorámica. Para ello se podrían obtener imágenes solapadas de la escena completa con distintos niveles de zoom y de alguna manera unirlas todas de forma que la calidad de la imagen panorámica aumentase.

Otra idea para aumentar la calidad de la imagen panorámica es utilizar la posición de la cámara para conocer qué tipo de transformación se le puede aplicar a la imagen sin necesidad de saberla imagen anterior, de forma que cada posición de la cámara resulta en una transformación conocida que da lugar a una imagen panorámica de mayor calidad.

Para el trabajo futuro también se recomienda buscar una manera de optimizar el código, en especial el de la segunda fase, con tal de que ésta se pueda realizar en tiempo real.

Una idea para llevar a cabo lo comentado en el párrafo anterior es utilizar la posición de la cámara para saber dónde insertar la imagen en la panorámica, o por lo menos para limitar el área donde se buscan puntos característicos en la imagen panorámica.

REFERENCIAS

[1] Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer vision and image understanding*, *110*(3), 346-359.

[2] Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381-395.

[3] Rong, W., Chen, H., Liu, J., Xu, Y., & Haeusler, R. (2009, November). Mosaicing of microscope images based on surf. In *Image and Vision Computing New Zealand, 2009. IVCNZ'09. 24th International Conference* (pp. 271-275). IEEE.

[4] Azzari, P., Di Stefano, L., & Bevilacqua, A. (2005, September). An effective real-time mosaicing algorithm apt to detect motion through background subtraction using a PTZ camera. In *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on* (pp. 511-516). IEEE.

[5] Brown, M., & Lowe, D. G. (2003, October). Recognising panoramas. In *ICCV*(Vol. 3, p. 1218).

[6] Yue, Z., Zhou, S. K., & Chellappa, R. (2004, May). Robust two-camera tracking using homography. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on* (Vol. 3, pp. iii-1). IEEE.

[7] Benhimane, S., & Malis, E. (2007). Homography-based 2d visual tracking and servoing. *The International Journal of Robotics Research*, *26*(7), 661-676.

[8] Grimson, W. E. L., Stauffer, C., Romano, R., & Lee, L. (1998, June). Using adaptive tracking to classify and monitor activities in a site. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on* (pp. 22-29). IEEE.

[9] Agapito, L., Hayman, E., & Reid, I. (2001). Self-calibration of rotating and zooming cameras. *International Journal of Computer Vision*, *45*(2), 107-127.

[10] Piccardi, M. (2004, October). Background subtraction techniques: a review. In*Systems, man and cybernetics, 2004 IEEE international conference on* (Vol. 4, pp. 3099-3104). IEEE.

[11] Wren, C. R., Azarbayejani, A., Darrell, T., & Pentland, A. P. (1997). Pfinder: Real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *19*(7), 780-785.

[12] Lo, B. P. L., & Velastin, S. A. (2001). Automatic congestion detection system for underground platforms. In *Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on* (pp. 158-161). IEEE.

[13] Bevilacqua, A., & Azzari, P. (2006, November). High-quality real time motion detection using ptz cameras. In *Video and Signal Based Surveillance, 2006. AVSS'06. IEEE International Conference on* (pp. 23-23). IEEE.

[14] Winkelman, F., & Patras, I. (2004, October). Online globally consistent mosaicing using an efficient representation. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on* (Vol. 4, pp. 3116-3121). IEEE.

[15] Hayman, E., & Eklundh, J. O. (2003, October). Statistical background subtraction for a mobile observer. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on* (pp. 67-74). IEEE.

[16] Bartoli, A., Dalal, N., Bose, B., & Horaud, R. (2002, December). From video sequences to motion panoramas. In *Motion and Video Computing, 2002. Proceedings. Workshop on* (pp. 201-207). IEEE.

[17] San Miguel, J. C., Bescós, J., Martínez, J. M., & García, Á. (2008, May). DiVA: a Distributed Video Analysis framework applied to video-surveillance systems. In *Image Analysis for Multimedia Interactive Services, 2008. WIAMIS'08. Ninth International Workshop on* (pp. 207-210). IEEE.

[18] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints.*International journal of computer vision*, *60*(2), 91-110.

[19] Herrero Martín, S., Bescós, J. (2009, April). Análisis comparativo de técnicas de segmentación de secuencias de video basadas en el modelado del fondo.